Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

Reviews

Machine Learning and Computer Vision Techniques in Self-driving Cars

[1]Saja Alaam Talib
Control and Systems Engineering Department / Computer Engineering Branch
University of Technology
Baghdad, Iraq
sajaalaam@yahoo.com

ARTICLEINFO
Article History
Received: 16/05/2024
Accepted:21/06/2024
Published: 30/11/2024
This is an open-access article under the CC BY 4.0 license:

ABSTRACT

This study explores the remarkable advancements in self-driving vehicles achieved through the application of computer vision and machine learning techniques. We examine various algorithms designed for critical functions, such as object detection, image segmentation, behavior prediction, and adaptive learning, which are all integral components of autonomous driving systems. Our research highlights key performance metrics, emphasizing accuracy, efficiency, and safety. Simulated environments and real-world testing are essential for validating the effectiveness of these methodologies. Our findings underscore the transformative potential of self-driving technology in enhancing transportation safety and its far-reaching effect on numerous industries. Notably, self-driving cars demonstrate the ability to reduce traffic accidents and improve traffic flow, which can lead to substantial economic and social benefits. Moreover, we discuss future research avenues, including the enhancement of system robustness and safety measures, the improvement of human–AI interaction, and the utilization of edge computing and edge AI. We also address the ethical and regulatory challenges associated with the widespread adoption of autonomous vehicles.

Our comprehensive analysis indicates that self-driving technology is poised to revolutionize the transportation sector, offering safer, more efficient, and more accessible mobility solutions. As technology continues to evolve, ongoing research and development will be crucial in overcoming current limitations and realizing the full potential of autonomous driving systems.

Keywords: Autonomous Vehicle, Computer Vision, Machine Learning, Path Planning, Simulation Testing.

List key index terms here. No more than 5.

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

1. INTRODUCTION

Self-driving cars represent a transformative leap in transportation technology, with the aim of revolutionizing mobility by reducing or eliminating the need for human intervention in driving tasks. The current landscape of self-driving cars showcases remarkable advancements, with numerous companies and paper institutions actively involved in the development and testing of autonomous vehicle systems. Central to this progress is the integration of computer vision and machine learning technologies. Computer vision enables vehicles to perceive and interpret their surroundings by analyzing visual data from cameras, LiDAR, and other sensors. Concurrently, machine learning algorithms play a crucial role in empowering autonomous vehicles to make intelligent decisions based on the interpreted data, thereby enhancing their ability to navigate safely and efficiently in various environments. Consequently, the fusion of computer vision and machine learning holds immense importance in advancing the capabilities of autonomous vehicles and driving them closer to widespread adoption [1].

This integration empowers self-driving cars to recognize and respond to various elements in their environment, including other vehicles, pedestrians, cyclists, traffic signs, and road markings. Through the utilization of computer vision techniques, autonomous vehicles can accurately detect and track objects, estimate their positions and velocities, and anticipate their future movements. Furthermore, machine learning algorithms enable these vehicles to learn from experience and adapt their behavior to different driving scenarios, thereby refining their decision-making capabilities over time [2].
As technology continues to evolve, researchers and engineers are exploring innovative approaches to enhance the performance, reliability, and safety of self-driving cars. Advancements in deep learning, sensor technology, and data fusion are driving new breakthroughs in perception, planning, and control algorithms. Additionally, efforts are underway to address challenges, such as robustness in adverse weather conditions, interaction with human-driven vehicles, and regulatory and ethical considerations.

Against this backdrop, this study aims to investigate the latest advancements in self-driving cars utilizing computer vision and machine learning. It delves into state-of-the-art techniques, methodologies, and challenges in perception, decision-making, and autonomous navigation. By elucidating the synergistic relationship between computer vision and machine learning in autonomous driving, this study seeks to contribute to the ongoing discourse on the future of transportation and pave the way for safer, more efficient, and more accessible mobility solutions [3]. Figure 1 illustrates the architecture of a self-driving car system. The system consists of several main components, including sensors (such as cameras, LiDAR, and radar), computer vision, machine learning, and planning and control. Sensors collect data from the environment, computer vision systems analyze these data to recognize objects and obstacles, and machine learning algorithms are used to make intelligent decisions based on this information. Finally, control systems implement these decisions to drive the car safely.

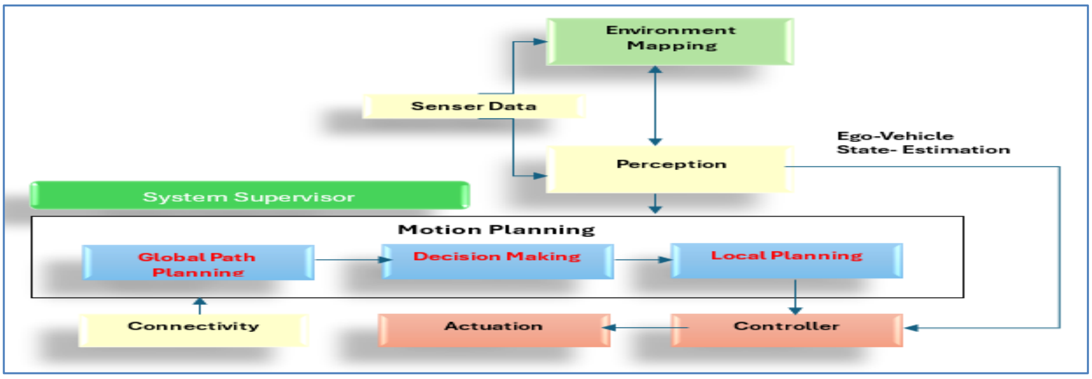

Fig. 1. Self-driving car system architecture

1.1 Problem of the Work

Despite remarkable advancements in computer vision and machine learning technologies, self-driving cars still face challenges related to robustness, reliability, and real-time decision-making. This work aims to address these challenges by investigating novel algorithms and methodologies to enhance the perception, navigation, and decision-making capabilities of autonomous vehicles in complex and dynamic environments.

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

1.2 Relevance of the Work

It lies in its potential to revolutionize transportation systems worldwide. By overcoming the current limitations of self-driving cars, such as safety concerns and adaptability to diverse environments, this work can pave the way for widespread adoption of autonomous vehicles. Consequently, it can lead to numerous benefits, including reduced traffic congestion, less accidents, increased accessibility for individuals with disabilities, and improved overall efficiency in transportation networks.

1.3 Objectives of the Work

- to investigate state-of-the-art computer vision and machine learning techniques applicable to self-driving cars
- to identify challenges and limitations in the current implementation of self-driving car systems

2. LITERATURE REVIEW

2.1 Overview of Self-Driving Car Technologies

Self-driving car technologies encompass a diverse array of methodologies and systems aimed at enabling vehicles to operate autonomously, free from human intervention. Several key technologies are pivotal in the development of autonomous vehicles [4]:

- Sensor systems: Self-driving cars rely on an array of sensors, including LiDAR, radar, cameras, and ultrasonic sensors, to perceive and interpret their surroundings. These sensors furnish vital data about the environment, including the detection of other vehicles, pedestrians, road signs, and obstacles.
- Computer vision: Computer vision techniques empower vehicles to comprehend and interpret visual information captured by cameras. Tasks such as object detection, image segmentation, and optical flow analysis are common in self-driving car systems, aiding in the identification of objects, lanes, and traffic signals. A depiction of self-driving cars utilizing computer vision is illustrated in Figure 1.
- Machine learning: Machine learning algorithms serve as the backbone for enabling autonomous vehicles to make intelligent decisions grounded in sensor data. Supervised learning, reinforcement learning, and deep learning methodologies are leveraged to train models for diverse tasks, including behavior prediction, path planning, and decision-making, as depicted in Figure 2.
- Localization and mapping: Localization techniques, such as GPS and simultaneous localization and mapping (SLAM), empower self-driving cars to ascertain their precise location and generate real-time maps of their surroundings. These maps are indispensable for navigation and route planning.
- Control systems: Control algorithms dictate the actions of the vehicle, encompassing steering, acceleration, and braking, to navigate safely and efficiently. Commonly employed control strategies in autonomous vehicles include model predictive control (MPC), proportional–integral–derivative control, and trajectory optimization.

In summary, the integration of these technologies equips self-driving cars with the ability to perceive their environment, make informed decisions, and navigate complex scenarios autonomously. Ongoing research and development endeavors persist in advancing these technologies, bringing us closer to the widespread adoption of self-driving cars [5]. Figure 2 illustrates the autonomous vehicle system block diagram. The main components include perception sensors, computer vision algorithms, machine learning techniques, planning and control systems, and communication and connectivity systems. The figure shows how these components interact with each other to collect information, analyze it, and make appropriate decisions to implement self-driving operations effectively.

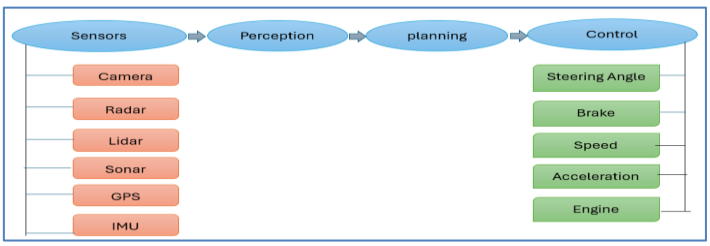

Fig. 2. Autonomous vehicle system block diagram

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

TABLE I. Comparison of Self-Driving Car Technologies

Method	Limitations	Gaps	Challenges	Advantages/Disadvantages
Sensor Systems	High cost of LiDAR, limited range of cameras	Integration of multiple sensors	Data fusion, real-time processing	Advantages: Provides comprehensive environmental data Disadvantages: Expensive, requires complex integration
Computer Vision	Sensitivity to lighting and weather conditions	Real-time processing of visual data	Object detection accuracy	Advantages: Enables visual perception Disadvantages: Affected by environmental conditions
Machine Learning	Requires large datasets for training, black-box nature	Real-time decision-making	Model interpretability	Advantages: Adaptive learning, improved decision-making Disadvantages: Requires extensive data, complex models
Localization and Mapping	GPS signal loss, SLAM computationally expensive	Accurate real-time localization	Handling dynamic environments	Advantages: Provides accurate positioning Disadvantages: GPS signal dependency, high computational load
Control Systems	Requires precise tuning, can be affected by sensor inaccuracies	Real-time control	Ensuring safety and reliability	Advantages: Provides precise vehicle control Disadvantages: Sensitive to sensor errors, requires fine-tuning

2.2 Role of Computer Vision

Computer vision plays a pivotal role in enabling vehicles to interpret and understand their surroundings in the domain of autonomous driving. Its importance lies in its capacity to extract meaningful information from visual data captured by onboard cameras and other imaging sensors. Here are key aspects underscoring the importance of computer vision [6]:

- Object detection and recognition: Computer vision algorithms facilitate the detection and recognition of various objects in the vehicle's environment, such as pedestrians, vehicles, cyclists, traffic signs, and lane markings. By accurately identifying and categorizing these objects, self-driving cars can make informed decisions to navigate safely and efficiently.
- Scene understanding: Computer vision techniques enable vehicles to comprehend complex scenes by analyzing the spatial relationships between objects, estimating distances, and understanding the context of the environment. This capability is essential for detecting potential hazards, predicting the behavior of surrounding entities, and planning appropriate driving maneuvers.
- Semantic segmentation: Computer vision algorithms segment images into meaningful regions, assigning semantic labels to pixels based on their corresponding object classes or categories. Semantic segmentation aids in understanding the layout of the scene, distinguishing between different road elements, and providing rich contextual information for decision-making processes.
- Optical flow analysis: Computer vision methods for optical flow estimation enable vehicles to perceive motion patterns in the environment, such as the movement of vehicles, pedestrians, and other dynamic objects. By tracking motion trajectories over time, self-driving cars can anticipate the future positions of objects and predict potential collision scenarios.
- Environmental perception: Computer vision enables vehicles to perceive a wide range of environmental factors, including lighting conditions, weather conditions, road surface characteristics, and traffic dynamics. This comprehensive perception allows autonomous vehicles to adapt their behavior, accordingly, ensuring robust performance across diverse driving scenarios.

In summary, computer vision serves as the eyes of self-driving cars, empowering them with the ability to observe, analyze, and understand their surroundings in real time. By leveraging computer vision technologies, autonomous vehicles can achieve higher levels of perception, situational awareness, and ultimately, safer, and more reliable autonomous driving experiences [7]. Figure 3 clarifies how computer vision works for self-driving car systems. Images and video are captured by cameras mounted on the car, and these data are then processed by computer vision algorithms to recognize objects and obstacles and determine their locations and speeds. This information helps understand the surrounding scene and make appropriate driving decisions.

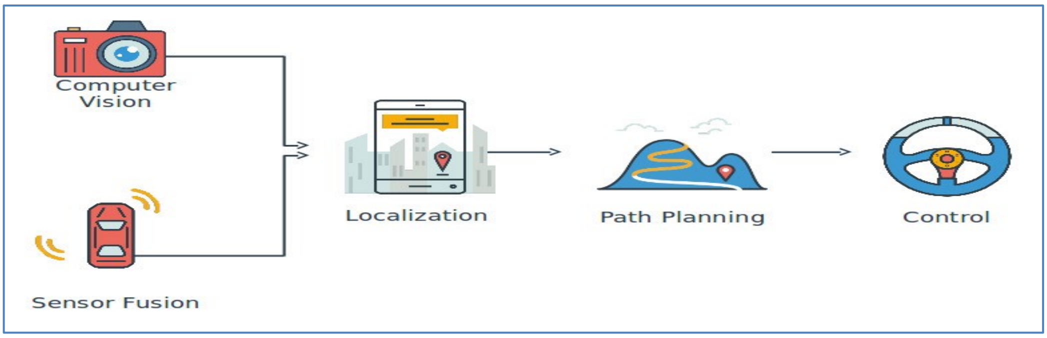

Fig.3. How computer vision works for self-driving car systems

2.3 Machine Learning in Autonomous Driving

Machine learning algorithms play a crucial role in enhancing decision-making and enabling adaptive behavior in self-driving cars. Here is an examination of how these algorithms contribute to various aspects of autonomous driving [8]:

- Behavior prediction: Machine learning models analyze sensor data to predict the behavior of other road users, such as pedestrians, cyclists, and other vehicles. By learning from historical data and real-time observations, these models can anticipate future trajectories and intentions, allowing the self-driving car to make proactive decisions to avoid potential collisions or conflicts. Figure 3 below illustrates the advantages and disadvantages of self-driving vehicles.
- Path planning: Machine learning techniques aid in generating optimal paths and trajectories for autonomous vehicles. By considering factors such as traffic conditions, road geometry, and dynamic obstacles, these algorithms can determine the safest and most efficient routes to navigate from the vehicle's current position to its destination. Reinforcement learning approaches further enable self-driving cars to adapt their navigation strategies based on feedback from the environment.
- Decision-making under uncertainty: Machine learning enables self-driving cars to make decisions in uncertain and complex situations. Bayesian inference methods, probabilistic modeling, and uncertainty estimation techniques allow autonomous vehicles to assess the reliability of sensor data, account for uncertainty in predictions, and make robust decisions even in challenging conditions, such as adverse weather or sensor failures.
- Adaptive learning: Machine learning algorithms facilitate continuous improvement and adaptation of autonomous driving systems over time. By leveraging feedback from real-world driving experiences and user interactions, these algorithms can update and refine their models to capture the nuances of driving behavior, road conditions, and environmental dynamics. Online learning and transfer learning techniques further enhance the scalability and generalization capabilities of self-driving car systems.

Overall, machine learning empowers self-driving cars with the ability to learn from data, adapt to changing environments, and make intelligent decisions autonomously. By harnessing the power of machine learning, autonomous driving systems can achieve higher levels of safety, efficiency, and reliability, paving the way for the widespread adoption of self-driving technology [9]. Figure 4 illustrates machine learning block diagram. The diagram includes the stages of data collection, data processing, training machine learning models, and testing and evaluating these models. This process allows the system to learn from past data and improve its performance in making decisions and planning paths.

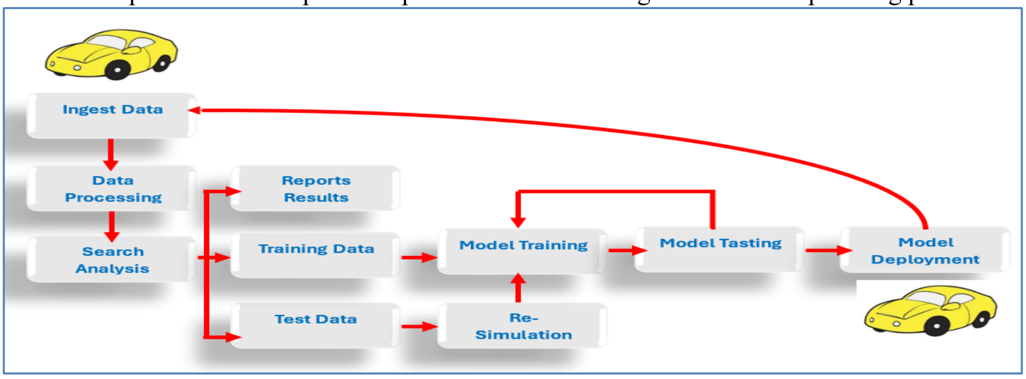

Fig. 4. Machine learning block diagram

2.4 Challenges and Limitations

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

The application of computer vision and machine learning in self-driving cars faces several challenges and limitations, which include the following [10]:

- Robustness to variability: Computer vision algorithms may struggle to generalize across diverse environmental conditions, such as changes in lighting, weather, and road infrastructure. Ensuring robustness and reliability in different scenarios remains a considerable challenge for autonomous driving systems.
- Data annotation and labeling: Training machine learning models for self-driving cars requires large, annotated datasets, which can be costly and time consuming to create. Furthermore, ensuring accurate labeling of complex scenes with multiple objects and interactions poses a considerable challenge in dataset preparation.
- Real-time processing: Real-time processing of high-dimensional sensor data, such as images and LiDAR point clouds, imposes computational constraints on self-driving car systems. Achieving low-latency inference while maintaining high accuracy is a technical challenge, particularly for resource-constrained embedded platforms.
- Safety and liability: Ensuring the safety of self-driving cars and addressing liability concerns in the event of accidents remain considerable challenges. Machine learning models may encounter edge cases and unforeseen scenarios where their behavior is unpredictable or potentially unsafe, raising ethical and regulatory challenges for deployment.
- Interpretability and explainability: Machine learning models used in autonomous driving systems often lack interpretability and explainability, making it challenging to understand their decision-making processes. Interpretable AI techniques are crucial for building trust and transparency in self-driving car technology, especially in safety-critical applications.
- Integration with traditional methods: Integrating computer vision and machine learning with traditional sensor fusion and control algorithms poses integration challenges. Harmonizing different components and ensuring seamless interaction between perception, decision-making, and control subsystems is essential for the overall performance of self-driving car systems.

Addressing these challenges and limitations requires interdisciplinary research efforts spanning computer vision, machine learning, robotics, and automotive engineering. By tackling these issues, researchers aim to accelerate the development and deployment of safe, reliable, and efficient self-driving car technology for real-world applications [8].

3. DEVELOPMENT OF COMPUTER VISION ALGORITHMS

3.1 Object Detection

Object detection is a fundamental task in computer vision for enabling self-driving cars to perceive and understand their surroundings. Several techniques have been developed for detecting and recognizing objects in the vehicle’s surroundings, including the following [11]:

- Convolutional neural networks (CNNs): CNNs have emerged as the dominant approach for object detection due to their ability to learn hierarchical features directly from raw pixel data. Architectures such as region-based CNNs (R-CNN), fast R-CNN, and faster R-CNN utilize CNNs for feature extraction and region proposal generation, followed by region-wise classification and bounding box regression to detect objects. Figure 5 clarifies the CNNs in self-driving car systems. CNNs process images captured by cameras and identify objects and obstacles with high accuracy. The figure shows how these networks consist of multiple layers that extract features and identify patterns that are important for driving decisions.

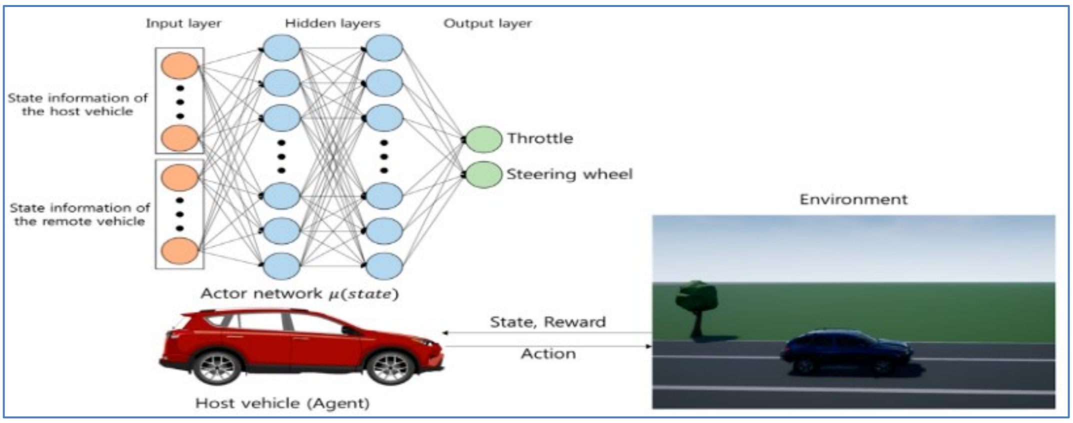

Fig. 5. Self-driving car with CNNs

- Single-shot multibox detector (SSD): SSD is a real-time object detection method that directly predicts object bounding boxes and class probabilities from feature maps at multiple scales. By incorporating convolutional layers with different aspect ratios, SSD achieves high detection accuracy and efficiency.
- You Only Look Once (YOLO): YOLO is another real-time object detection algorithm that formulates object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. YOLO achieves fast inference speeds by directly predicting bounding box coordinates and class probabilities in a single pass through the network. Figure 6 below clarifies object detection by using the YOLO technique. This algorithm divides the image into a grid and identifies objects in each grid cell. The figure shows how the YOLO algorithm can recognize different objects, such as cars, pedestrians, and bicycles, in driving scenes.

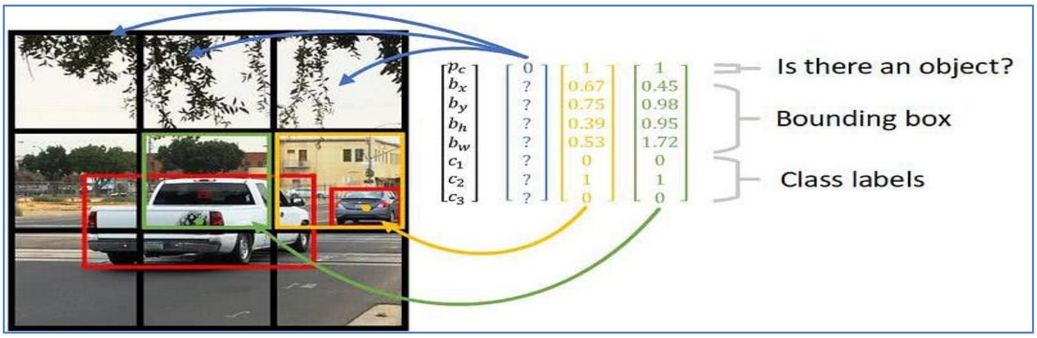

Fig. 6. Object detection using YOLO

- Efficient Det: Efficient Det is a recent advancement in object detection that introduces efficient model architectures and compound scaling techniques to achieve state-of-the-art performance with few parameters and computational resources. By optimizing network architectures and scaling strategies, Efficient Det achieves a good trade-off between accuracy and efficiency.

These techniques for object detection enable self-driving cars to detect and localize various objects in their surroundings, including vehicles, pedestrians, cyclists, traffic signs, and other relevant entities. By accurately identifying objects and estimating their positions, self-driving cars can make informed decisions and navigate safely and efficiently in diverse driving scenarios. Ongoing research continues to improve the performance, speed, and robustness of object detection algorithms, bringing us closer to the realization of fully autonomous driving systems [4].

TABLE II. Comparison of Object Detection Techniques

Method	Limitations	Gaps	Challenges	Advantages/Disadvantages
Convolutional Neural Networks (CNNs)	High computational cost, requires large, annotated datasets	Real-time detection and processing	Handling diverse and complex scenes	Advantages: High accuracy, hierarchical feature learning Disadvantages: Computationally intensive, data hungry
Region-based CNNs (R-CNN, Fast R-CNN, Faster R-CNN)	Slow inference speed, complex training process	Efficient region proposal generation	Balancing speed and accuracy	Advantages: Accurate object detection Disadvantages: Slow inference, complex training
Single Shot Multibox Detector (SSD)	Reduced accuracy for small objects	Maintaining high accuracy across scales	Balancing speed and detection accuracy	Advantages: Real-time detection, simple architecture Disadvantages: Reduced accuracy for small objects
You Only Look Once (YOLO)	Struggles with detecting close and small objects	Improving detection accuracy	Achieving high accuracy with speed	Advantages: Real-time performance, fast inference Disadvantages: Reduced accuracy for small/close objects
Efficient Det	Trade-off between model complexity and performance	Achieving state-of-the-art performance	Optimizing network architecture	Advantages: Good balance of accuracy and efficiency Disadvantages: Requires careful tuning for optimal performance

3.2 Image Segmentation

Image segmentation is a critical task in computer vision that involves partitioning an image into semantically meaningful regions or segments. In the context of self-driving cars, image segmentation is used to identify and distinguish different

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

elements on the road, such as lanes, vehicles, pedestrians, and obstacles. Several methods for image segmentation have been developed, including the following:

- Semantic segmentation: Semantic segmentation assigns a class label to each pixel in an image, effectively segmenting the image into regions corresponding to different object categories. CNNs, particularly fully convolutional networks, have shown remarkable success in semantic segmentation tasks by learning to predict pixel-wise class probabilities. Architectures such as U-Net, SegNet, and DeepLab utilize variations of CNNs with encoder–decoder structures to produce dense semantic segmentation masks [12].
- Instance segmentation: Instance segmentation goes a step further than semantic segmentation by not only labeling each pixel with a class label but also distinguishing individual object instances within the same class. Mask R-CNN is a popular instance segmentation algorithm that extends Faster R-CNN by adding a parallel branch for predicting segmentation masks alongside bounding box coordinates and class probabilities. This enables accurate segmentation of object instances while simultaneously detecting and classifying objects [12].
- Panoptic segmentation: Panoptic segmentation aims to unify semantic and instance segmentation by segmenting stuff (e.g., road, sky) and things (e.g., vehicles, pedestrians) in an image. Panoptic segmentation methods typically combine semantic segmentation with instance segmentation techniques to produce a comprehensive segmentation map that covers all elements in the scene. Models such as panoptic FPN and UPSNet achieve panoptic segmentation by integrating semantic segmentation and instance segmentation predictions in a unified framework [9].
- Lane segmentation: Lane segmentation is a specialized form of image segmentation that focuses on detecting and delineating lane markings on the road. Lane segmentation algorithms typically employ computer vision techniques, such as edge detection, color-based segmentation, and line fitting, to identify lane boundaries. Deep learning-based approaches, such as LaneNet and ENet, utilize CNNs to predict lane markings from road images directly, achieving robust and accurate lane segmentation results [9].

These methods for image segmentation enable self-driving cars to perceive and understand the road environment accurately by segmenting images into distinct regions corresponding to different elements on the road. By effectively segmenting images, self-driving cars can extract rich contextual information for navigation, path planning, and decision-making, ultimately enhancing their autonomy and safety on the road [2]. Figure 7 illustrates the different image segmentation techniques used in autonomous vehicle systems. These techniques include semantic segmentation, spot segmentation, panoramic segmentation, and line segmentation. The figure illustrates how these techniques help identify and distinguish different elements in a driving scene, such as roads, sidewalks, obstacles, and traffic signs.

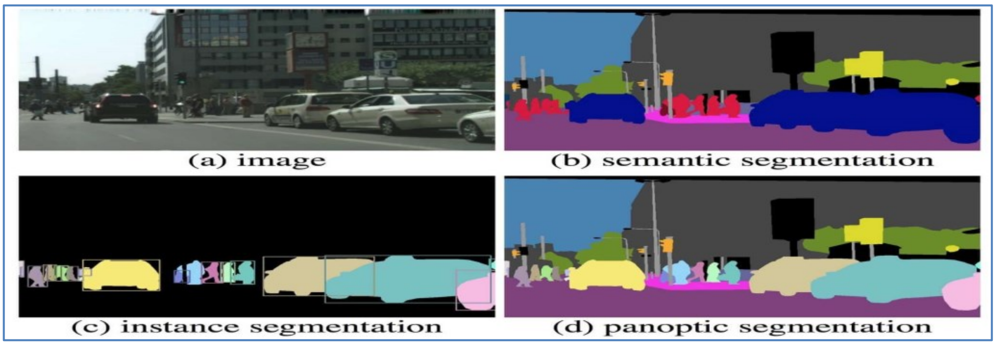

Fig. 7. Image segmentation methods

TABLE III. Comparison of Image Segmentation Methods

Method	Limitations	Gaps	Challenges	Advantages/Disadvantages
Semantic Segmentation	Struggles with differentiating object instances	Fine-grained segmentation of instancing	High computational cost	Advantages: Detailed pixel-wise classification Disadvantages: Cannot distinguish between instances of the same class
Instance Segmentation	Complex and computationally intensive	Integration with real-time processing	Balancing accuracy and speed	Advantages: Differentiates between instances within the same class Disadvantages: Computationally heavy, requires large datasets

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

Panoptic Segmentation	High complexity, combines challenges of semantic and instance segmentation	Real-time unified segmentation framework	Efficiently combining semantic and instance output	Advantages: Comprehensive scene understanding Disadvantages: Complex to implement, high computational cost
Lane Segmentation	Limited to specific features (lane markings), may struggle in poor visibility conditions	Robustness in diverse environmental conditions	Accurate detection in various lighting and weather conditions	Advantages: Crucial for navigation and path planning Disadvantages: Specialized, may fail in adverse conditions

3.3 Path Planning

Path planning is a crucial component of autonomous driving systems; it involves determining optimal paths for vehicles based on interpreted data from computer vision and other perception sensors. Several algorithms and techniques are used for path planning in self-driving cars:

- An Algorithm: The A algorithm is a widely used path planning algorithm that efficiently searches for the shortest path between a start and a goal state in a graph or grid-based environment. By iteratively expanding the search space and evaluating candidate paths based on a combination of heuristic cost estimates and actual path costs, A can find optimal paths while minimizing computational overhead [13].
- Dijkstra's Algorithm: Dijkstra's algorithm is another classic path planning algorithm that finds the shortest path between nodes in a weighted graph. In contrast to A, Dijkstra's algorithm explores all possible paths from the start node to all other nodes in the graph, considering the cumulative cost of each path. While Dijkstra's algorithm guarantees finding the shortest path, it may be computationally expensive for large-scale environments [13].
- Rapidly exploring random trees (RRT): RRT is a probabilistic sampling-based algorithm commonly used for motion planning in high-dimensional continuous spaces. RRT incrementally grows a tree structure by randomly sampling and extending tree branches toward unexplored areas of the configuration space. RRT-based methods, such as RRT and RRT-Connect, efficiently explore the state space and generate feasible paths for self-driving cars in complex and dynamic environments [6].
- MPC: MPC is a control-based approach to path planning that formulates the path planning problem as a finite-horizon optimization task. By predicting the future evolution of the vehicle's state and optimizing control inputs over a finite time horizon, MPC generates dynamically feasible and collision-free trajectories while accounting for vehicle dynamics, constraints, and environmental uncertainties [6].
- Hybrid approaches: Hybrid approaches combine elements of geometric and sampling-based methods to achieve efficient and robust path planning in complex environments. Techniques such as probabilistic roadmaps and hybrid A integrate global and local planning strategies to generate smooth and dynamically feasible paths while considering static and dynamic obstacles [6].

These algorithms for path planning enable self-driving cars to navigate safely and efficiently in diverse environments by generating optimal paths based on interpreted data from computer vision and other perception sensors. By selecting suitable path planning algorithms and integrating them into autonomous driving systems, self-driving cars can navigate complex scenarios while adhering to traffic rules, avoiding collisions, and reaching their destinations efficiently [8].

TABLE IV. Comparison of Path Planning Methods

Method	Limitations	Gaps	Challenges	Advantages/Disadvantages
An Algorithm	Can be computationally expensive in large environments	Efficient heuristics for large-scale use	Balancing optimality and computational efficiency	Advantages: Finds optimal paths, efficient for smaller environments Disadvantages: Computationally intensive for large environments
Dijkstra's Algorithm	Computationally expensive, especially in large-scale environments	Scalability to large graphs	Handling large-scale environments	Advantages: Guarantees shortest path Disadvantages: Explores all paths, high computational cost
Rapidly exploring Random Trees (RRT)	May produce suboptimal paths, requires collision checking	Generating optimal paths	Ensuring smoothness and optimality	Advantages: Efficient in high-dimensional spaces, good for dynamic environments Disadvantages: Can generate nonoptimal paths, requires tuning
Model Predictive Control (MPC)	Requires accurate vehicle and environment models, computationally intensive	Real-time implementation	Handling dynamic constraints and uncertainties	Advantages: Generates feasible, collision-free trajectories Disadvantages: Requires accurate models, high computational load
Hybrid Approaches	Complexity in implementation and tuning	Integration of global and local planning	Balancing efficiency and robustness	Advantages: Combines strengths of multiple methods Disadvantages: Complex to implement and tune

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

4. MACHINE LEARNING FOR DECISION-MAKING

4.1 Behavior Prediction

Behavior prediction is a critical aspect of decision-making in self-driving cars because it involves forecasting the future actions and trajectories of other vehicles, pedestrians, and obstacles in the environment. Various machine learning models and techniques have been developed for behavior prediction, including the following [14]:

- Recurrent neural networks (RNNs): RNNs are a class of neural networks designed to process sequential data, making them well-suited for modeling temporal dependencies in behavior prediction tasks. Long short-term memory networks and gated recurrent units are popular variants of RNNs that can capture long-term dependencies and dynamics in the behavior of road users over time.
- Trajectory prediction: Trajectory prediction models aim to forecast the future trajectories of vehicles, pedestrians, and other dynamic objects in the scene. These models typically learn from historical trajectory data and sensor observations to predict future motion patterns and intentions. Techniques such as sequence-to-sequence models, attention mechanisms, and mixture density networks are commonly used for trajectory prediction tasks.
- Interaction-aware models: Interaction-aware models consider the interactions between multiple agents in the scene when predicting behavior. Graph-based models, such as graph neural networks, model the spatial and semantic relationships between agents and encode social cues and norms into the prediction process. These models capture complex interactions, such as merging, lane changing, and yielding, between vehicles and pedestrians in traffic scenarios.
- Uncertainty estimation: Uncertainty estimation techniques enable behavior prediction models to quantify and propagate uncertainty in their predictions. Bayesian neural networks, ensemble methods, and Monte Carlo dropout techniques are used to estimate uncertainty in trajectory predictions, allowing self-driving cars to make more informed decisions and plan robust trajectories in uncertain environments.
- Multimodal prediction: Multimodal prediction models generate multiple plausible future trajectories or behavior for each agent in the scene, capturing the inherent uncertainty and variability in human behavior. Generative models, such as variational autoencoders and generative adversarial networks, produce diverse sets of future predictions, enabling self-driving cars to anticipate and adapt to different possible outcomes.

By leveraging these models for behavior prediction, self-driving cars can anticipate and react to the actions of other road users, thereby enhancing safety, efficiency, and comfort in autonomous driving scenarios. Ongoing research in behavior prediction aims to improve the accuracy, robustness, and interpretability of predictive models, advancing the capabilities of self-driving car systems in real-world environments [15].

TABLE V. Comparison of Behavior Prediction Methods

Method	Limitations	Gaps	Challenges	Advantages/Disadvantages
Recurrent Neural Networks (RNNs)	Struggle with long sequences, can be computationally expensive	Capturing long-term dependencies	Handling variable-length sequences	Advantages: Good for sequential data, captures temporal dependencies Disadvantages: Difficult to train, may struggle with long-term dependencies
Trajectory Prediction	Requires large datasets, can be sensitive to noise in data	Robust real-time trajectory prediction	Balancing accuracy and computational cost	Advantages: Predicts future motion patterns Disadvantages: Sensitive to data quality, high computational cost
Interaction-aware Models	Complex implementation, requires modeling of interactions between multiple agents	Robust interaction modeling	Handling multiple, complex interactions	Advantages: Captures interactions between agents Disadvantages: Complex, computationally intensive
Uncertainty Estimation	Can be computationally intensive, requires advanced techniques for uncertainty quantification	Accurate uncertainty modeling	Balancing model accuracy and uncertainty estimation	Advantages: Quantifies prediction uncertainty Disadvantages: Computationally expensive, complex methods
Multimodal Prediction	Generating diverse but plausible trajectories can be challenging	Comprehensive future behavior modeling	Ensuring diversity and plausibility	Advantages: Captures multiple possible outcomes Disadvantages: Difficult to ensure plausibility, computationally intensive

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

4.2 Adaptive Learning

Adaptive learning enables self-driving cars to improve their performance continuously and adapt to diverse driving scenarios through experience and feedback. Several approaches and techniques have been developed to facilitate adaptive learning in autonomous driving systems, including the following [16]:

- Reinforcement learning (RL): Reinforcement learning is a machine learning paradigm where agents learn optimal behavior by interacting with their environment and receiving feedback in the form of rewards or penalties. In the context of self-driving cars, RL algorithms can learn to navigate and make decisions in complex environments by trial and error. Techniques such as deep Q-networks, policy gradient methods, and actor–critic algorithms enable self-driving cars to learn adaptive behavior, such as lane-keeping, merging, and yielding, from interactions with simulated or real-world driving environments. Figure 8 shows how the A* algorithm is used to plan the path of a self-driving car. The algorithm is based on finding the optimal path from the starting point to the target destination, considering the obstacles in the way. The figure shows the network of nodes and how the paths are evaluated based on the movement costs to determine the optimal path.

Fig. 8. Reinforcement learning in machine learning

- Online learning: Online learning algorithms enable self-driving cars to update their models incrementally and adapt to changes in the environment in real time. By continuously collecting data and updating model parameters, online learning allows autonomous vehicles to learn from new driving scenarios and adapt their behavior accordingly. Algorithms such as online gradient descent, online random forests, and online support vector machines facilitate adaptive learning in self-driving cars by efficiently processing streaming data and updating model parameters on-the-fly.
- Transfer learning: Transfer learning leverages knowledge acquired from previous tasks or domains to accelerate learning and adaptation in new environments. Pretrained models and features extracted from large-scale datasets or simulated environments can be fine-tuned or transferred to specific self-driving car tasks, such as object detection, path planning, and behavior prediction. Transfer learning enables self-driving cars to leverage existing knowledge and adapt more quickly to new driving scenarios with limited data.
- Metalearning: Metalearning algorithms enable self-driving cars to learn how to learn by efficiently adapting to new tasks or environments with minimal training data. Metalearning approaches, such as model-agnostic metalearning (MAML) and Reptile, optimize model parameters to enable rapid adaptation and generalization across diverse driving scenarios. By learning to quickly adapt to new environments, self-driving cars can improve their performance and safety in novel and challenging situations.

By incorporating adaptive learning approaches into autonomous driving systems, self-driving cars can continually evolve and improve their capabilities, robustness, and safety in response to changing environmental conditions, traffic patterns, and user preferences. Adaptive learning plays a vital role in advancing the development and deployment of self-driving car technology, enabling autonomous vehicles to navigate complex real-world scenarios with confidence and reliability [17].

TABLE VI. Comparison of Adaptive Learning Methods

Method	Limitations	Gaps	Challenges	Advantages/Disadvantages
Reinforcement Learning (RL)	Requires extensive training, can be sample inefficient	Efficient exploration strategies	Balancing exploration and exploitation	Advantages: Learns optimal behaviors through interaction Disadvantages: Sample inefficient, requires extensive training

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

Online Learning	May struggle with concept drift, requires continuous data flow	Handling nonstationary environments	Real-time data processing and adaptation	Advantages: Adapts to changes in real-time Disadvantages: Sensitive to noisy data, may struggle with concept drift
Transfer Learning	Requires relevant source domains, may not always be applicable	Efficiently transferring knowledge	Ensuring transferability and relevance	Advantages: Accelerates learning with limited data Disadvantages: Dependent on the relevance of source domain
Metalearning	High complexity, may require large computational resources	Efficient adaptation to new tasks	Rapidly adapting with minimal data	Advantages: Quickly adapts to new environments Disadvantages: Computationally intensive, complex algorithms

5. EVALUATION AND VALIDATION

5.1 Performance Metrics

In assessing the effectiveness of algorithms developed for self-driving cars, comprehensive performance metrics capturing accuracy, efficiency, and safety must be established. Commonly used metrics include the following [18]:

- Accuracy:

- Object detection: Precision, recall, and F1-score gauge the accuracy of object detection algorithms in correctly identifying and localizing objects.
- Trajectory prediction: Mean average displacement error and mean average speed error assess the accuracy of trajectory prediction models in forecasting motion paths.
- Path planning: Metrics such as path completion rate and time-to-goal evaluate the accuracy of path planning algorithms in generating optimal trajectories.

- Efficiency:

- Computational efficiency: Inference time, memory usage, and energy consumption measure the computational efficiency of algorithms on resource-constrained hardware.
- Planning time: This metric measures the time taken by path planning algorithms to generate feasible paths, ensuring timely response in dynamic scenarios.

- Safety:

- Collision avoidance: This metric assesses the ability of self-driving cars to detect and avoid collisions with other objects in the environment.
- Violations: This metric quantifies safety by evaluating adherence to traffic regulations and behavior norms.

5.2 Simulation and Real-World Testing

Simulated and real-world testing are pivotal for validating the effectiveness and reliability of self-driving car methods [19]:

- Simulation testing:

- Cost effective: provides a cost-effective means of testing and evaluating algorithms in controlled and repeatable scenarios
- Scalability: allows for the generation of diverse driving scenarios to comprehensively evaluate algorithm performance
- Rapid iteration: facilitates quick experimentation and refinement of algorithms without real-world deployment constraints

- Real-world testing:

- Realistic conditions: validates algorithm performance under realistic driving conditions, including unpredictable factors, such as weather and traffic
- Safety validation: ensures compliance with regulatory standards and user expectations regarding algorithm safety and robustness
- User experience: provides insights into user experience and acceptance of autonomous driving technology, guiding user-centric solutions

By integrating simulation and real-world testing approaches, researchers can thoroughly validate proposed methods for self-driving cars, advancing the development and deployment of autonomous vehicle technology [20].

6. DISCUSSION

6.1 Technologies Used

- Online learning: Online learning algorithms enable self-driving cars to update their models gradually and adapt to changes in the environment in real time. These algorithms include online stochastic learning and online support

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

vector machines, which facilitate adaptive learning by processing streaming data and updating model parameters on the fly.

- Transfer learning: Transfer learning takes advantage of knowledge acquired from previous tasks or fields to accelerate learning and adaptation in new environments. Pretrained models and features extracted from large datasets or simulated environments can be adapted for specific tasks, such as object detection, path planning, and behavior prediction.
- Metalearning: Metalearning algorithms enable self-driving cars to learn how to learn by efficiently adapting to new tasks or environments with minimal training data. Metalearning methods, such as MAML and Reptile, are used to optimize the model parameters to enable rapid adaptation and generalization across diverse driving scenarios.

6.2 Problems and Limitations

- Facing adverse weather conditions: Weather conditions, such as rain and snow, are major challenges for autonomous driving systems. These conditions require enhanced sensing and data processing technologies to ensure efficient system performance.
- Interaction with human cars: Seamless interaction between self-driving cars and human-driven cars is crucial. This requires algorithms that can predict the behavior of other drivers and make appropriate decisions in real time.
- Regulatory and ethical considerations: Self-driving cars face regulatory and ethical challenges related to safety and accountability in the event of accidents. Companies need to comply with regulatory standards and ensure public acceptance of this technology.

6.3 Proposed Solutions

- Improved sensing and deep learning: By developing new sensing technologies and improving the algorithms used in deep learning, the ability of self-driving cars to better see and interact with their environments can be improved.
- Continuous learning and adaptation: Using continuous learning to enable cars to adapt to changes in the environment and continuously improve performance based on new data collected during operation.
- Experimentation and simulation: Simulation and real-world testing are important to verify the effectiveness and reliability of autonomous driving methods. Simulation provides a cost-effective environment for experimentation and improvement, while real-world testing ensures performance in real conditions.

7. CONCLUSION

The integration of computer vision and machine learning technologies has propelled remarkable advancements in the development of self-driving cars. These technologies enable autonomous vehicles to perceive their environment, make informed decisions, and navigate safely and efficiently through diverse conditions. The synergy between computer vision and machine learning has unlocked new potentials for enhancing the performance, reliability, and scalability of self-driving car systems. Data-driven approaches and continuous learning mechanisms allow autonomous vehicles to adapt to evolving driving conditions, foresee potential hazards, and optimize their driving behavior, thereby ensuring passenger safety and comfort.

Despite the promising future of self-driving cars, several challenges remain, including regulatory hurdles, ethical concerns, and technical limitations. Overcoming these challenges require collaborative efforts from researchers, engineers, policymakers, and stakeholders across various sectors. Looking ahead, the focus must be shifted toward addressing these challenges and further refining the capabilities of self-driving cars. By advancing innovation and leveraging the power of computer vision and machine learning, we can anticipate a future where autonomous vehicles integrate seamlessly with traditional transportation systems, ushering in a new era of mobility and convenience for all.

7.1 Improving Computer Vision Techniques

- Exclusively enable object detection: Developing new technologies, such as CNN and YOLO networks, to detect objects in various details accurately and quickly
- Semantic segmentation techniques: Improving the use of semantic segmentation techniques to understand complex environments and navigate with precise information

7.2 Develop Learning Algorithms

- Behavior analysis and prediction: Improving large predictive models of vehicle and pedestrian behavior using reinforcement learning and other learning techniques

- Modification: Developing modification algorithms that can play a key role in new data and real-world motion experiments

7.3 Improving Planning and Navigation Systems

- Development of strategic planning algorithms: Using algorithms, such as MPC and RRT, to improve critical planning of routes in complex and dynamic environments
- Elimination of the current traditional and modern: Integrating vision and machine learning technologies with traditional sensing and control systems for improved performance

7.4 Improve Liquidity and Reliability

- Inclusion of expanded scenarios: Expanding liquidity cycles to include diverse, complex, and electronically reliable scenarios
- Interaction with the human car: Studying the interaction of self-driving cars with conventional cars and the safety and driving effectiveness of shared driving

7.5 Dealing with Animal and Organizational Challenges

- Causes of issues: Developing ways to deal with issues related to autonomous vehicle decisions in emergency situations
- Regulatory compliance: Working with stakeholders on standards and policies that support safe technology deployment

7.6 Improving Advanced Computing Techniques

- Using cloud computing: From edge computing technologies to the comprehensive internet and reliance on cloud networks
- AI-on-chassis integration: Integrating AI techniques into electronics to improve computational performance and speed

7.7 Collaboration, Search for Knowledge

- Collaboration among innovative innovators: Encouraging collaboration between universities to exchange knowledge for effective and potential solutions
- Publication: Open to support open data facilitate data banking between sectors and developers

References

[1] L. William and Z. Weihua, "Self-driving cars and the city: Effects on sprawl, energy consumption, and housing affordability". Regional Science and Urban Economics, Vol.81, 2020, Self-driving cars and the city: Effects on sprawl, energy consumption, and housing affordability - ScienceDirect

[2] J. Miller, "Self-Driving Car Technology's Benefits, Potential Risks, and Solutions", theenergycollective.com. Archived from the original on 8 May 2015. Retrieved 2024., 2014.

[3] C. Henry, B. N. Amitai, and G. Jeffrey, “Self-Driving Cars: The Impact on People with Disabilities”, Ruderman Family Foundation/Securing America's Future Energy, 2017.

[4] L. Maxim, “Deep Reinforcement Learning Hands-On”, 2nd edition, 2020. https://www.packtpub.com/product/deep-reinforcement-learning-hands-on-second-edition/9781838826994

[5] S. S. Richard and B. G. Andrew, “Reinforcement Learning: An Introduction”, The MIT Press m, Cambridge, Massachusetts, London, England, 2nd edition ,2018, Pages 2-4 https://mitpress.mit.edu/books/reinforcement-learning-second-edition

Iraqi Journal for Computers and Informatics
Information Technology and Communications University
Vol. 50, No. 2, 2024, pp. 75-89
DOI: https://doi.org/10.25195/ijci.v50i2.498
Print ISSN: 2313-190X, Online ISSN: 2520-4912

[6] Friedman and R. Steichen, “Light-Duty Automotive Technology, Carbon Dioxide Emissions, and Fuel Economy Trends: 1975 Through 2016”, National service center for Environmental publications, 2016. http://www.epa.gov/otaq/fetrends.htm - Search (bing.com)

[7] Z. Jianfeng, L. Bodong and Ch. Qiuxia, "The key technology toward the self-driving car". International Journal of Intelligent Unmanned Systems, Vol., No.1, 2018, pp: 2–20. doi:10.1108/IJIUS-08-2017-0008. ISSN 2049-6427 .

[8] Z. Wentao; M. Jun et al, “Vehicle detection in driving simulation using extreme learning machine". Neurocomputing, Vol.128, 2014, pp.160–165. doi:10.1016/j.neucom.2013.05.052.

[9] S. M. L. Hazel, T. Araz,"Algorithmic Decision-Making in AVs: Understanding Ethical and Technical Concerns for Smart Cities". Sustainability journal, 2019.

[10] W. S. Bryant, “Proximity-Driven Liability,” Georgetown Law Journal, Vol. 102, 2014.

[11] P. W. Stephen, Ch. Jesse and et al “The Potential Regulatory Challenges of Increasingly Autonomous Motor Vehicles,” Santa Clara Law Review, Vol. 52, 2012, pp. 1423–1502.

[12] W. S. Bryant, “Automated Vehicles Are Probably Legal in the United States”, Stanford, Calif.: Stanford Law School, Center for Internet, and Society, 2014. Automated Vehicles Are Probably Legal in the United States by Bryant Walker Smith :: SSRN

[13] W.S. Bryant, “Managing Autonomous Transportation Demand,” Santa Clara Law Review, Vol. 52, No. 4, 2012, pp. 1401–1422. Managing Autonomous Transportation Demand by Bryant Walker Smith:: SSRN

[14] James Davidson and Marcus, “Challenges and Opportunities of Road Vehicle Automation,” presented at Transportation Research Board Annual Meeting, San Jose, Calif., July 16, 2017a.

[15] Society of Automobile Engineers, Seat Belt Comfort, Fit, and Convenience—Truck and Bus, standard J1834, 2007.

[16] Society of Automobile Engineers, On-Road Automated Vehicle Standards Committee, Draft Information Report J 3016: Taxonomy and Definitions, unpublished, 2015.

[17] Society of Automobile Engineers, Safety and Human Factors Steering Committee, Adaptive Cruise Control (ACC) Operating Characteristics and User Interface, standard J2399, December 2016.

[18] R. Marincik and P. Moyle, “Human Factors in Forward Collision Warning Systems: Operating Characteristics and User Interface Requirements”, standard J2400, 2018.

[19] J2400A (WIP) Human Factors in Forward Collision Warning Systems: Operating Characteristics and User Interface Requirements - SAE International

[20] M. Graham and D. Lain, “A Simulated Autonomous Car”, thesis, University of Edinburgh, 2012.

[21] Knight, "The Future of Self-driving Cars". MIT Technology Review, 2023.